

DESARROLLO DE APLICACIÓN EN ANDROID PARA CONTROL REMOTO DE DATOS DE UN CONTROLADOR LÓGICO PROGRAMABLE

DEVELOPMENT OF AN ANDROID APPLICATION FOR REMOTE CONTROL OF PROGRAMABLE LOGIC CONTROLLER DATA

Carlos Andrés Angulo Martínez

Tecnológico Nacional de México en Celaya, México
14031626@itcelaya.edu.mx

María Teresa Parra Bárcenas

Tecnológico Nacional de México en Celaya, México
14031806@itcelaya.edu.mx

Juan Pablo Aguilera Álvarez

Tecnológico Nacional de México en Celaya, México
juan.aguilera@itcelaya.edu.mx

Ángel Cárdenas León

Tecnológico Nacional de México en Celaya, México
angel.cardenas@gmail.com

Recepción: 18/septiembre/2019

Aceptación: 3/octubre/2019

Resumen

En este proyecto se presenta el desarrollo de una aplicación para dispositivos móviles compatible con el sistema operativo Android, que trabaje como una Interfaz Humano-Máquina (HMI), para el monitoreo y control remoto de datos, utilizando un Controlador Lógico Programable (PLC). La aplicación se desarrolló en un Entorno de Desarrollo Integrado (IDE) con lenguaje de programación Java, ejecutando pruebas para una Computadora Personal (PC), hasta lograr la aplicación definitiva desarrollada en Android Studio, la comunicación entre el PLC y el dispositivo móvil se realizó con la librería Moka7, que ofrece soporte de conexión vía Ethernet, finalmente, se adaptó a ser una conexión inalámbrica entre los dos dispositivos. El prototipo incluye una interfaz gráfica donde es posible agregar los controles e indicadores que desee el usuario y configurar cada uno de los elementos. Este

sistema es capaz de leer y escribir 5 tipos de variables: entero, doble entero, real, booleano y string.

Palabra(s) Clave: Aplicación en Android, Comunicación inalámbrica de datos, Controlador Lógico Programable, Java, Moka7.

Abstract

An application for remote monitoring and control of data, such as a Human Machine Interface (HMI), operating with a Programmable Logic Controller (PLC), it was developed for mobile devices that work with Android operating system. An Integrated Development Environment (IDE) was used for application development, made use of the Java programming language with a library called Moka7, which offers Ethernet connection to achieve communication between the PLC and the mobile device, then become a wireless connection between the two devices. In a Personal Computer (PC) was performed the test programs, and finally, in Android Studio achieved the final application. A system capable of reading and writing five types of variables: boolean, integer, double integer, real and string. It has a menu where it is possible to add the corresponding controls and indicators, as well as, from the graphical interface it is possible to add the controls and indicators that the user desires.

Keywords: *Android Application, Wireless data communication, Programmable Logic Controller, Java, Moka7.*

1. Introducción

Algunos estudios han demostrado que la enseñanza virtual del uso de los PLC es muy útil en el entorno académico enfocado a la capacitación profesional [Cáceres & Amaya, 2016]. Asimismo, en la mayoría de los sistemas de control de PLC es necesario comunicarlos con dispositivos periféricos como equipos "inteligentes" y las HMI, que están presentes en la industria [Technology, 2019], son el medio con que el usuario puede comunicarse con una máquina, equipo o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo. Por lo general, suelen ser fáciles de entender y accionar [Solórzano, 2014]. Se sabe que por lo

menos el 45% de las empresas poseen una HMI, por la que acceden a los datos desde una distancia de 1 metro [Castro et al., 2014]. E.g., se tiene una aplicación que controla de forma remota un proceso industrial de un PLC, envía datos a un microprocesador y este se conecta a un dispositivo que manda datos por bluetooth y los datos se envían desde un teléfono inteligente [Bonifaz & Solórzano, 2018]. E.g., es la aplicación móvil que alerta automáticamente cuando alguna máquina de preparación de cable deja de operar, o por alguna falla, al responsable de estas, es por lo que la supervisión de máquinas de manera remota está siendo utilizada cada vez más en la industria en general [Esquivel, 2015].

Hoy en día, los servicios de Internet son ampliamente utilizados en la industria y es muy común encontrar comunicación Ethernet en una planta industrial, facilitan la comunicación entre distintos dispositivos, e.g., PLC's aumentando el rendimiento y las posibilidades de control [Prado, 2010]. Existen diferentes sistemas de comunicación Ethernet orientados a procesos de control industrial como Ethernet/IP [ODVA, 2012] o Profinet [Profibus, 2012], que permiten el control remoto de entradas y salidas, y que, junto con unas estrictas características de temporización, garantizan un correcto sistema de control [Rosado-Muñoz et al., 2012]. El tener acceso a una red de internet para un monitoreo y control remoto de algún proceso a través de un navegador web, puede mejorar la flexibilidad de trabajo, permitiendo al personal controlar equipamiento desde otro lugar [Shyr et al., 2013]. La aplicación ofrece el monitoreo de las variables desde el dispositivo móvil, considerando que, actualmente el uso de estos es bastante común y es más práctico que una computadora, además son de los aparatos de mayor consumo, puesto que forman parte del estilo de vida de un número creciente de personas en todo el mundo [Cruz-Sotelo et al., 2013]. Las actividades que se realizaron durante el proyecto son los que se mencionan a continuación: aprender el software TIA Portal y las bases de programación del lenguaje Java, revisar las ventajas y desventajas de los IDE, seleccionar uno de ellos para la aplicación en PC, determinar la funcionalidad del IDE, con el código desarrollado en Java comunicar la PC con el PLC, leer y escribir datos desde la PC al PLC, familiarizarse con Android Studio para importar el código de la comunicación, finalmente, diseñar una Interfaz Gráfica de Usuario (GUI).

En esta investigación, se desea tener acceso, monitoreo y control remoto de las variables del PLC S7-1200, a través de un dispositivo móvil con sistema operativo Android. Realizar la programación en el lenguaje Java, y hacer uso de la librería Moka7, que permite la comunicación entre ambos dispositivos. Primero desde la PC, posteriormente pasarse al dispositivo móvil diseñando una interfaz amigable con el usuario. Las variables de interés son del tipo: entero, doble entero, real, booleano y string.

2. Métodos

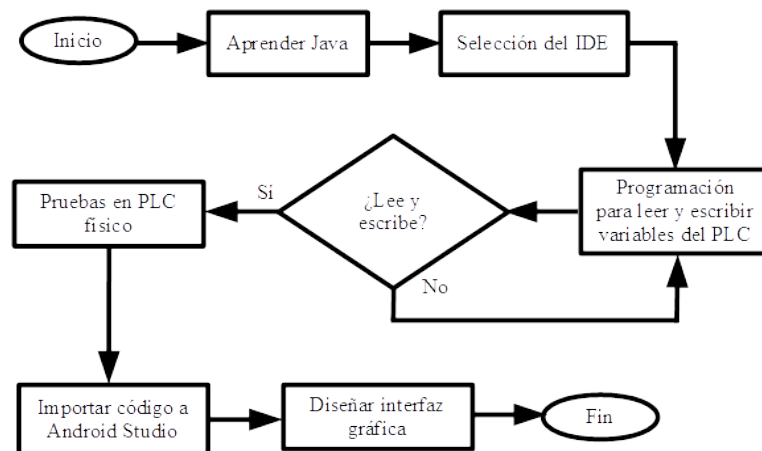
Se utilizaron controladores lógicos programables de la marca SIEMENS S7-1200, modelo 1211C AC/DC/RLY, un repetidor inalámbrico Steren COM-818, bancos de prueba para visualizar las salidas del autómata y activar las entradas digitales mediante botones y una librería de código abierto llamada Moka7 para lograr la comunicación entre el PLC y el dispositivo móvil. Se emplearon los siguientes programas: Android Studio para programar en dispositivos con sistema operativo Android; IntelliJ IDEA para programar en PC, ambos son entornos de desarrollo para programar en Java; y Totally Integrated Automation (TIA) Portal en el que se programan los PLC SIEMENS.

El procedimiento utilizado para desarrollar la aplicación es el siguiente:

1. Se estudiaron los lenguajes de programación: Java, permitiendo conocer su sintaxis, comandos, etc., siendo la base para la programación de la aplicación y el lenguaje escalera utilizando el software TIA Portal, donde se programan los PLC SIEMENS.
2. Existen algunos IDE's: Eclipse, NetBeans, IntelliJ IDEA y Android Studio, para elegir uno de ellos se buscaron sus ventajas y desventajas. La ventaja principal encontrada en el tercer entorno, de los ya mencionados, es que permite importar el código a Android Studio, por lo que se eligió IntelliJ IDEA para la programación en PC y Android Studio para el desarrollo de la aplicación en dispositivos Android.
3. Se inició la programación en el entorno elegido en el paso anterior, buscando leer y escribir un tipo de dato del PLC, una vez lograda la comunicación, se

- hizo para cada uno de los cuatro restantes y que el usuario tenga la opción de elegir la variable que desee leer o escribir.
4. Lograda la comunicación, resta hacer pruebas en el PLC físico, corroborar que el código funcione de manera correcta, de no ser así, hacer las correcciones necesarias.
 5. Se continúa por importar el código a Android Studio, con una interfaz similar a la que se tiene en PC e implementando una comunicación inalámbrica.
 6. Utilizando el lenguaje kotlin, se trabajó en un código en el que se puede crear y borrar un led, por medio de botones, con la propiedad de arrastrarse a cualquier posición del panel, limitado por los bordes de la pantalla.
 7. Se crearon dos menús, el primero se encuentra en la parte superior de la pantalla, permite conectarse/detener la comunicación con el PLC, configuración, información de la aplicación, a su vez se cuenta con un botón de añadir elemento en el que aparece el segundo menú que es desplegable.
 8. Con el código realizado en el paso 6, fue posible implementar los elementos de tipo control, indicador y contenido extra. Se encuentran presentes en el menú desplegable a disposición del usuario.

En la figura 1, se muestra el diagrama de flujo del método utilizado para la realización de la aplicación.

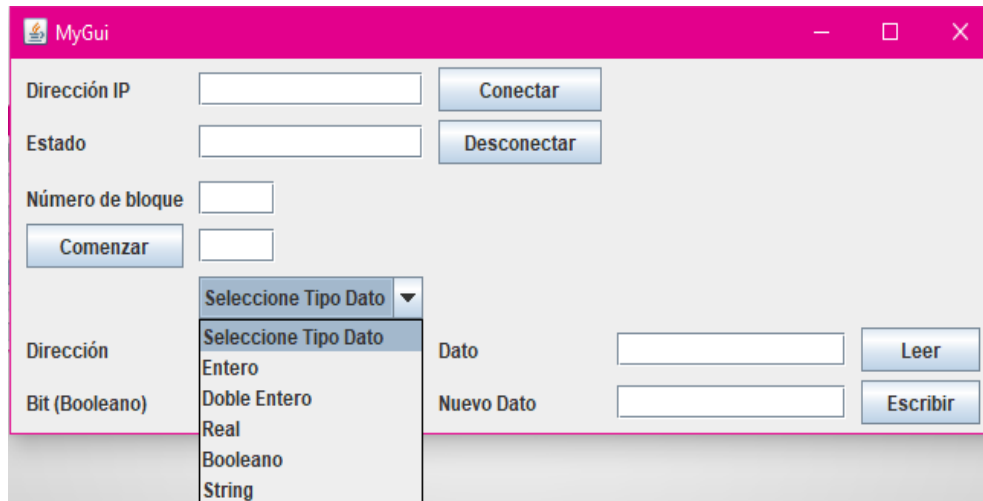


Fuente: Elaboración propia.

Figura 1 Método de la aplicación móvil.

3. Resultados

Se logró leer y escribir cada uno de los tipos de datos de interés, por lo que en la figura 2, se ilustra la interfaz en la que el usuario es capaz de elegir el tipo de dato que desea.



Fuente: Elaboración propia.

Figura 2 Interfaz de comunicación (software libre).

Para la comunicación se define la dirección IP del PLC y el número de bloque. Dentro del bloque se encuentran las variables, una vez identificados los parámetros de la variable a controlar, se llenan los tres campos: tipo de dato, dirección y bit (sólo para datos booleanos). Finalmente, se tiene la opción de leer donde se muestra el estado actual de la variable y escribir donde se modifica su valor.

Se utilizó la misma librería Moka7 para los dispositivos Android, iniciando por una interfaz similar a la obtenida en PC, se consiguieron resultados favorables al realizar la transmisión de datos de forma inalámbrica mediante un módem. Se creó una versión preliminar para empezar a trabajar en la aplicación del dispositivo móvil, como se visualiza en la figura 3.

Finalmente, se diseñaron los elementos empezando por los controles (botón push/switch, perilla, deslizador, cuadro de control), indicadores (led, tanque, termómetro, deslizador, cuadro indicador) y contenido extra (cuadro de texto y regla) con el PLC, ver figura 4.



Fuente: Elaboración propia.

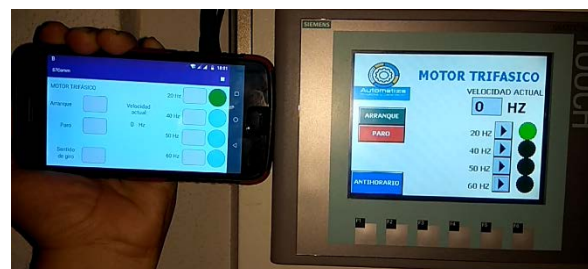
Figura 3 Primera versión de la interfaz (Android).



Fuente: Elaboración propia.

Figura 4 Menú con todos los elementos.

Se realizaron una serie de pruebas en la empresa Automatiza con la aplicación, que consistieron en el encendido y apagado de un motor trifásico de corriente alterna donde es posible cambiar entre las velocidades de 20, 40, 50 y 60 hertz y el sentido de giro. Se hicieron las pruebas tanto en la aplicación, como en la HMI que se encuentra en la empresa, se representa en la figura 5.



Fuente: Elaboración propia.

Figura 5 Pruebas en empresa Automatiza.

4. Discusión

Al pasar el código que se tenía para la comunicación en PC al software Android Studio, cambió un poco la programación, ya que en IntelliJ IDEA los programas se ejecutan bajo una función principal, de esta manera se trabajó la parte de la comunicación. Y en Android Studio se manejan actividades, que se manejan como eventos. Cada actividad se crea con un archivo xml, que es el archivo de diseño en los que se agregan los objetos con los que el usuario ha de interactuar y editar los atributos de los objetos. Al inicio se tenía pensado dar soporte a sistemas Android versión 7 en adelante, sin embargo, como se observa en la figura 6, existe poco porcentaje de usuarios que utilizan Android 7 (37.1%) y desde Android 6 se cubre un mayor porcentaje de mercado, también, tras el desarrollo del programa se encontró que el código es funcional para la versión 6 y se optó por habilitar el uso de la aplicación para dichas versiones.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99.6%
4.2 Jelly Bean	17	98.1%
4.3 Jelly Bean	18	95.9%
4.4 KitKat	19	95.3%
5.0 Lollipop	21	85.0%
5.1 Lollipop	22	80.2%
6.0 Marshmallow	23	62.6%
7.0 Nougat	24	37.1%
7.1 Nougat	25	14.2%
8.0 Oreo	26	6.0%
8.1 Oreo	27	1.1%

Fuente: (Android Studio, junio 2019).

Figura 6 Porcentaje de usuarios por versiones Android.

En lo que respecta a la programación de la interfaz gráfica, se optó por utilizar el lenguaje de programación Kotlin, dado que ya es un lenguaje oficial para el desarrollo de aplicaciones Android (Club de Tecnología, 2017) y por las ventajas que se muestran en la tabla 1, se optó por utilizarse para la interfaz. Se cumplieron satisfactoriamente los objetivos, los cuales son:

- Acceder a las variables del PLC desde la computadora, comunicación y transferencia de variables por medio de una aplicación Android.
- El desarrollo de una GUI realizada en un IDE.
- Evitar que el usuario no se encuentre permanentemente cerca del proceso.

Tabla 1 Comparación de lenguajes de programación.

JAVA	KOTLIN
Permite asignar un "null" para cualquier variable, pero si se quiere hacer uso de alguna que tenga el valor "null", se lanza un error en tiempo de ejecución.	No es posible asignar "null" a una variable, y esto cumple para todos los tipos.
Se debe escribir más código.	Reduce en gran medida el código extra.
Únicamente se hacen llamadas a funciones Kotlin.	Está diseñado para interoperar con java.

Fuente: Club de Tecnología, 2017.

La aplicación es una alternativa de las HMI en las empresas, debido a que cuenta con los elementos básicos para monitorear y controlar las variables. El tiempo de respuesta de la aplicación, dependerá del proceso que se esté trabajando, los elementos que se estén utilizando y también el dispositivo móvil desde el que se esté monitoreando y/o controlando. Para concluir, la hipótesis es aceptada, ya que sí es posible hacer el control y monitoreo de datos de forma remota a partir del uso de un módem que se conecta al PLC, lo que permitió hacer la transferencia de hasta 5 tipos de variables (entero, doble entero, real, booleano, string), mediante una aplicación para dispositivos móviles Android. Se recomienda aprender el lenguaje Java y/o Kotlin, son la base para la programación, tanto de la comunicación, como de la interfaz gráfica. Mediante pequeños programas en los que se familiarice con la sintaxis y la programación orientada a objetos, y por consiguiente, desarrollar aplicaciones sencillas en Android. El primer paso, será conocer el entorno, dado que en él se manejan actividades que están divididas por un archivo xml y Java.

Como línea futura, la aplicación puede tener las siguientes mejoras:

- Persistencia de datos: Guardar el proyecto en el que se esté trabajando y acceder a él en el momento que se requiera. Para ello, es necesario investigar acerca de las bases de datos o archivos.
- Más espacio de trabajo: Por ahora, la aplicación cuenta con un espacio de trabajo limitado, más específicamente, se refiere a la pantalla principal del

dispositivo móvil que se esté utilizando, e.g., si es una tablet, tendrá el usuario más espacio de trabajo. Se pretende lograr mover la pantalla hacia abajo o hacia los lados, con el fin de tener más espacio para la realización de proyectos en la aplicación.

- Crear más elementos: Con la finalidad de ser una aplicación más completa y que el usuario tenga un mayor número de elementos a su disposición para monitorear o controlar las variables de su interés.
- Importar imágenes: Colocar cualquier imagen que se encuentre dentro del dispositivo para que la aplicación sea más interactiva.
- Dar soporte a más tipos de datos. Existen más tipos de datos con los que es posible tener comunicación entre la aplicación y el PLC, e.g., los de tipo palabra, doble palabra, de tiempo, etc.
- Cambiar de tamaño los elementos: Debido a que los dispositivos móviles tienen distintos tamaños de pantallas, se espera lograr hacer que los elementos tengan la opción de reducir o aumentar su tamaño, de tal forma que el usuario pueda adaptarlo como desee.
- Más variedad de colores al led: Este elemento sólo cuenta con un color, por lo que se espera darle más variedad de colores en un futuro, de tal modo que el usuario tenga la facilidad de elegir el color de su agrado.
- Descripción de cada elemento: Añadir el elemento al panel con su respectivo nombre y número de forma automática.

5. Bibliografía y Referencias

- [1] Cáceres, C. A., & Amaya, D. (2016). Desarrollo e interacción de un laboratorio virtual asistido y controlado por PLC. *Entre Ciencia e Ingeniería*, vol. 10, núm. 19, 9-15. ISSN # 1909-8367.
- [2] Esquivel, F. J. (2015). Aplicación móvil para el monitoreo de máquinas KOMAX 433 de preparación de cable en la industria automotriz. Tesis. Universidad Autónoma de Ciudad Juárez.
- [3] Bonifaz, A., & Solórzano, J. (2018). Diseño e implementación de una aplicación de monitoreo, control de velocidad y sentido de giro de un motor

- monofásico AC utilizando un smartphone con sistema Android. Ecuador. Tesis. Universidad Politécnica Salesiana.
- [4] Castro, J., Laguna, J., & Vargas, M. (2014). La aplicación de interfaces humano-máquina (HMI), en la industria Tijuanaense. In *Ciencias de la Ingeniería y Tecnología Handbook T-VI: Congreso Interdisciplinario de Cuerpos Académicos* (pp. 204-218). ECORFAN.
- [5] Club de Tecnología, (2017, septiembre). Kotlin vs Java, principales diferencias de los lenguajes oficiales de Android: www.clubdetecnologia.net/blog/2017/kotlin-vs-java-principales-diferencias-de-los-lenguajes-oficiales-de-android/.
- [6] Cruz-Sotelo, S. E., Benítez, S. O., Bovea, M. D., Soto, N. S., Ávila, H. F., & Salinas, W. E. A. (2013). Hábitos y prácticas de consumo de teléfonos celulares en México y España. *Revista Internacional de Contaminación Ambiental*, 29, 33-41.
- [7] ODVA (2012), <http://www.odva.org>
- [8] Prado, J. C. (2010). Ethernet Industrial: Modelos y conectividad en el ámbito de procesos industriales. Tesis. Universidad Nacional de La Plata. Facultad de Informática.
- [9] Profibus (2012), <http://www.profibus.com>
- [10] Rosado-Muñoz, A., Mjahad, A., Muñoz-Marí, J., & Terol-Tortosa, G. (2012). Web monitoring System and Gateway for Serial Communication PLC. Alemania.
- [11] Shyr, W. J., Su, T. J., & Lin, C. M. (2013). Development of remote monitoring and a control system based on PLC and webaccess for learning mechatronics. *International Journal of Advanced Robotic Systems*, 10 (2), 97. <https://doi.org/10.5772%2F55363>.
- [12] Solórzano H. (2014). Diseño e Implementación de Interfaz Gráfica de Usuario entre el Sistema de Control de Proceso T5554 y un PLC Siemens S7-300.
- [13] Technology, B. (2019). Top causes of PLC control system failure: <https://www.boultingtechnology.co.uk/news/view/top-causes-of-plc-control-system-failure/>.